BCSC01/0019/2021- JOHNSON KANYI WAMWEYA

ASSIGNMENT ONE

BCIT 3134: SOFTWARE PROJECT MANAGEMENT

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

a. Explain the evolution of software economics. (3 marks)

1. ***Conventional Phase (1960s-1970s)***

   During this period, software development was highly customized with unique tools and processes and often bundled with hardware. The cost was mainly driven by hardware and software was an added advantage. The focus was on achieving basic functionality, often with underachieved goals due lack of standardized practices. Software components included Fortran and PL/1. The quality of performance was poor, less great and was over budget and schedule

2. ***Transition Phase (19780s-1990s)***

   This era saw the introduction and use of various repeatable processes and off-the-shelf tools generally developed in high- level programming languages. The commercial software products and components include databases, networking along with operating systems, database management systems and graphical user interface. But due to increase in complexity, all the languages and technologies available were not enough for desired business performance

3. ***Modern Phase (2000s-Present)***

   The current phase emphasizes managed processes, IDEs with cloud computing, automation, DevOps practices which have become prevalent, aiming to enhance flexibility, speed, collaboration, reduced cost and enable scalability and continuous models in software development.

b. Draw the activity network and the critical path as well as estimate to determine the critical path and shortest time to complete the project (6 marks)

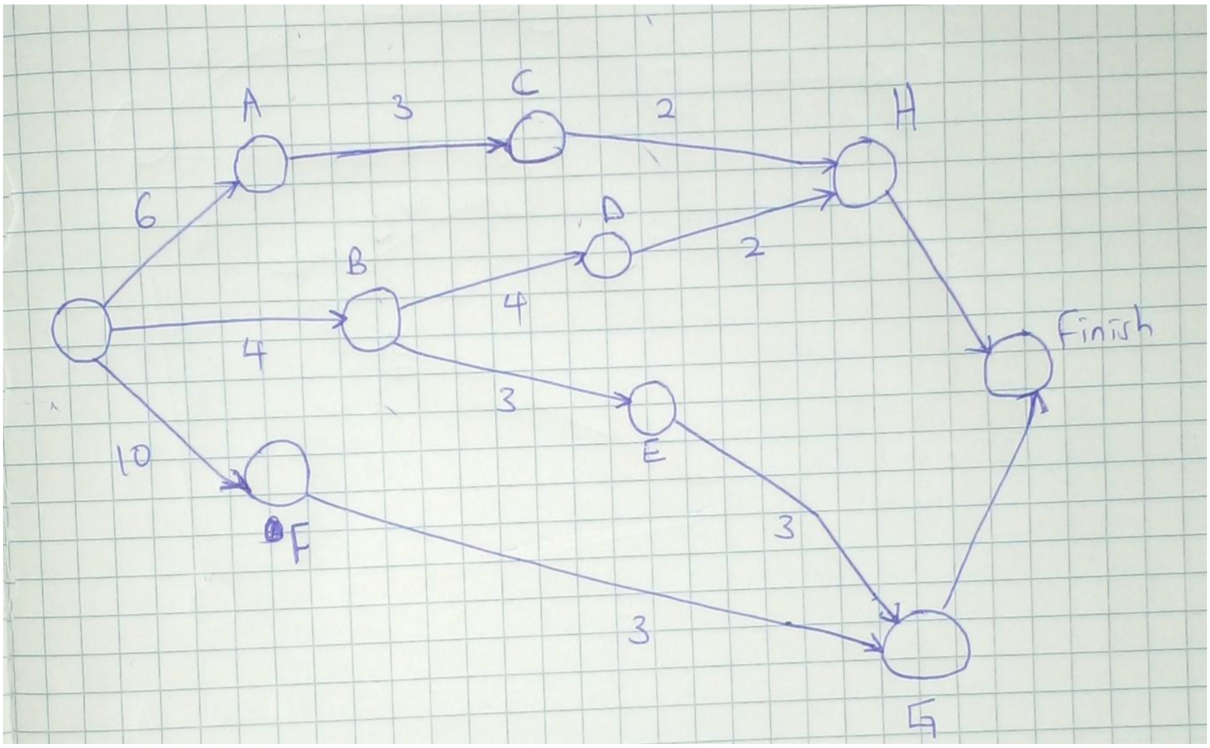
DURATION(WEEKS)	PRECEDENTS
6	-

4	**-**
3	**A**
4	**B**
3	**B**
10	**-**
3	**E, F**
2	**C, D**

The activities and their duration along with their precedents:

A- Hardware Selection (6 weeks)
B- Software Design (4 weeks)
C- Install Hardware (3 weeks, after activity A)
D- Test Software (3 weeks, after activity B)
E- File Take-on (3 weeks, after activity B)
F- Write User Manuals (10 weeks)
G- User Training (3 weeks, after activity E, F)
H- Install and Test (2 weeks, after C, D)

Activity Network Diagram:



Project Duration:

A → C → H : 6 + 3 + 2 = 11 weeks

B → D → H : 4 + 4 + 2 = 10 weeks

B → E → G : 4 + 3 + 3 = 10 weeks

F → G : 10 + 3 = 13 weeks

Critical path: F → G where it equals (10 + 3) weeks= 13 weeks

Hence the answer is ***13 weeks.***

c. A project is estimated to be finished in one year, using two programmers. However, due to a change in circumstances, it is desirable that the project be speeded up. A proposal is put forward to increase the number of programmers to six, thus allowing the project to be completed in 4 months.

Required:

(i) What would be the risk, if the project committee does not come up with any new proposal to cope the change in circumstances and stick to one-year plan despite of changes in circumstances (5 marks)

1. ***Increased Costs*** due to prolonged project duration due to extended resource usage and potential inflation
2. ***Stakeholder Dissatisfaction*** due to slower delivery leading to loss of confidence in the project team.
3. ***Competitive Disadvantage*** as competitors may release similar product sooner reducing the project's impact.
4. ***Resource Inefficiency*** as resources might be underutilized if the project takes longer than necessary.
5. ***Operational Inefficiencies*** as a slower delivery may delay operational benefits of the software affecting productibility and affordability.

(ii) What would be the risk, if the project committee allows the project to be completed in 4 months according to new proposal? (6 marks)

1. ***Higher Costs*** as hiring more programmers and speeding up the schedule typically increases project costs.

2. ***Team Coordination Challenges;*** managing a larger team in a shorter timeframe can result in communication overhead and coordination difficulties.
3. ***Team Burnout (Diminishes Team Productivity)*** due to increased number of programmers and speeding up the timeline.
4. ***Quality Compromise*** as rushing the project might lead to inadequate testing and quality assurance.
5. ***Increased Complexity*** as larger teams often introduce complexities in code integration and version control.
6. ***Poor Learning Curve;*** the new programmers might need time to understand the project affecting productivity.